AD-756 904

ADAPTIVE CONVERGENCE STUDIES. EXTENDED ARRAY EVALUATION PROGRAM

Aaron H. Booker, et al

Texas Instruments, Incorporated

Prepared for:

Advanced Research Projects Agency

30 April 1972

DISTRIBUTED BY:

NTIS

National Technical Information Service
U. S. DEPARTMENT OF COMMERCE
5285 Port Royal Road, Springfield Va. 22151

AFTAC Project No. VT/1705

ADAPTIVE CONVERGENCE STUDIES

SPECIAL REPORT NO. 1
EXTENDED ARRAY EVALUATION PROGRAM

Prepared by Aaron H. Booker, Chung-Yen Ong, and Thomas E. Barnard

T. W. Harley, Program Manager
Area Code 703, 836-3882 Ext. 300

TEXAS INSTRUMENTS INCORPORATED
Services Group
P.O. Box 5621
Dallas, Texas 75222

Contract No. F33657-71-C-0843
Amount of Contract: $511,580
Beginning 1 April 1971
Ending 31 March 1972

Prepared for

AIR FORCE TECHNICAL APPLICATIONS CENTER
Washington, D.C. 20333

Sponsored by

ADVANCED RESEARCH PROJECTS AGENCY
Nuclear Monitoring Research Office
ARPA Order No. 1714
ARPA Program Code No. 1F10

30 April 1972

Acknowledgement: This research was supported by the Advanced Research Projects Agency, Nuclear Monitoring Research Office, under Project VELA-UNIFORM, and accomplished under the technical direction of the Air Force Technical Applications Center under Contract No. F33657-71-C-0843.

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*

Texas Instruments Incorporated
Services Group
P.O. Box 5621, Dallas, Texas 75222

2a. REPORT SECURITY CLASSIFICATION

UNCLASSIFIED

2b. GROUP

3. REPORT TITLE

Adaptive Convergence Studies
Extended Array Evaluation Program Special Report No. 1

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

Special Report No. 1

5. AUTHOR(S) *(First name, middle initial, last name)*

Aaron H. Booker
Chung-Yen Ong

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
15 March 1972	~~38~~ 39	1

8a. CONTRACT OR GRANT NO.

Contract No. F33657-71-C-0843

b. PROJECT NO.

AFTAC Project No. VT/1705

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)

9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10. DISTRIBUTION STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

11. SUPPLEMENTARY NOTES

ARPA Order No. 1714

12. SPONSORING MILITARY ACTIVITY

Advanced Research Projects Agency
Nuclear Monitoring Research Office
Arlington, Virginia 22209

13. ABSTRACT

The purpose of this study is to investigate the ability of a modified adaptive algorithm to increase the convergence rate on weak noise components. A detailed mathematical derivation of the modified algorithm is presented. In addition, performance of the algorithm on synthetic data is evaluated. The algorithm is applied in both a single loop and a two loop mode. Both single-constraint and multiple-constraint modes of operation are investigated.

The results of this study indicate that the multiloop adaptive filtering algorithm can be used to speed the rate at which an adaptive filter set approaches optimality.

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Adaptive Convergence Study						
Multi-loop Adaptive Algorithm						
Single Constraint Algorithm						
Multiple Constraint Algorithm						

ABSTRACT

The purpose of this study is to investigate the ability of a modified adaptive algorithm to increase the convergence rate on weak noise components. A detailed mathematical derivation of the modified algorithm is presented. In addition, performance of the algorithm on synthetic data is evaluated. The algorithm is applied in both a single loop and a two loop mode. Both single-constraint and multiple-constraint modes of operation are investigated.

The results of this study indicate that the multiloop adaptive filtering algorithm can be used to speed the rate at which an adaptive filter set approaches optimality.

TABLE OF CONTENTS

LIST OF FIGURES

SECTION I

INTRODUCTION

The rate of convergence of an adaptive filter algorithm to a neighborhood of the optimum filter is approximated in terms of the eigenvalues corresponding to the principal components of some matrix closely associated with the noise matrix (Brennan, 1971). For the frequency-domain maximum-likelihood adaptive algorithm, the adaptive update equation is:

$$A^{t+1} = \left[I - 2\mu \left(I - \frac{VV^H}{V^H V} \right) X^t (X^H)^t \right] A^t ,$$

where X^t is the input data vector at the t^{th} iteration, V^H is the beamsteer filter, A is the conjugate transpose of the adaptive filter vector at the $(t+1)^{th}$ and t^{th} iterations, respectively, and μ is a real scalar quantity controlling the adaptation rate. The superscript H denotes conjugate transpose. The output of the adaptive filter at the t^{th} iteration is $(A^H)^t X^t$. If we substitute $\Phi = E(XX^H)$ for $X^t(X^H)^t$ in the update equation for this particular algorithm, the time constant τ_p is approximately

$$\frac{1}{2\mu\lambda_p}$$

for energy lying on the p^{th} orthonormalized eigenvector of the matrix

$$\left(I - \frac{VV^H}{V^H V} \right) \Phi \left(I - \frac{VV^H}{V^H V} \right)$$

where $\mu << 1/\lambda_p$. Here λ_p is the p^{th} eigenvalue of that matrix.

τ_p is the time constant for the p^{th} principal component in the sense that the portion of the excess filter output RMS

$$\sqrt{A^H \Phi A - (A^H \Phi A)_{minimum}}$$

associated with the p^{th} principal component of the matrix

$$\left(I - \frac{VV^H}{V^H V}\right) \Phi \left(I - \frac{VV^H}{V^H V}\right)$$

is reduced by a factor $e = 2.71828$ in approximately $1/(2\mu\lambda_p)$ iterations. Similarly, the amplitude

$$\left| a_p - (a_p)_{optimum} \right|$$

of the difference between the adaptive filter and the optimum maximum-likelihood filter along the p^{th} principal component is likewise reduced by a factor e in the same period of time. It is assumed that substitution of Φ for $X^t(X^t)^H$ does not change the answer much.*

Stability of the adaptive algorithm requires that

$$\mu < 1/\lambda_{max}$$

so that the time constant for weak components may be very great and μ cannot be chosen sufficiently large for practical effectiveness against very weak components. The purpose of the present study is to investigate the ability of modified adaptive algorithms to increase the convergence rate on weak noise components.

For the purpose of simplicity, the modified algorithms in the frequency domain will be discussed, although in the following section the modified algorithms are developed in both the time and frequency domains. Let there be a sequence of data vectors X^t at a specific frequency. These vectors

* This would be the case if successive data vectors were independent. In the dependent data vector case at hand, one might expect this to be true for μ very small.

are to be processed adaptively so as to preserve a signal $S = cV/(V^H V)$, where c is a complex scalar quantity and V is the conjugate transpose of the beamsteer filter V^H. One way to formulate this algorithm is

$$o_1^t = V^H X^t - (F_1^H)^t X^t ,$$

where o_1^t is the output of the first loop at time t, and $(F_1^H)^t$ is an adaptive filter applied to X^t to predict the complex-valued scalar quantity $V^H X^t$. If the constraint $(F_1^H)^t V = 0$ is imposed for all t, then it follows for the input $X^t = S = cV/(V^H V)$ that the output is

$$o_1^t = \frac{c\left[V^H V - (F_1^H)^t V\right]}{V^H V}$$

$$= \frac{cV^H V}{V^H V}$$

$$= c$$

so that the combined filter $V^H - F_1^H$ has a unit response to the signal. It is possible to consider a second loop or filter

$$o_2^t = o_1^t - (F_2^H)^t X^t$$

or in general

$$o_k^t = o_{k-1}^t - (F_k^H)^t X^t$$

where the signal is still preserved if the constraints

$$(F_k^H)^t V = 0$$

are required. If the filters are to operate independently, the optional additional constraints

$$(F_j^H)^t \, F_k^t = 0 \qquad (j \neq k)$$

need to be specified. The algorithm with only the constraint to preserve the signal will be termed the single-constraint mode algorithm, whereas for the algorithm with the additional constraints, the terminology multiple-constraint mode will be used. Heuristically, the second loop allows the filters F_k^H to concentrate on the smaller components of the noise because F_1^H has reduced the largest noise component. The following section will develop the algorithm mathematically, and then in the final section some results will be given to indicate the behavior of the algorithms on synthetic data.

SECTION II

MATHEMATICAL DEVELOPMENT OF THE TECHNIQUE

A. MULTILOOP ADAPTIVE FILTERING - FREQUENCY DOMAIN

1. Steepest Descent Algorithm with Complex Linear Constraints

Let f be a real-valued function of the real variables x_1, y_1, x_2, y_2, ..., x_n, y_n. Suppose that these variables are subject to the m complex constraint equations $g_k(x_1, y_1, x_2, y_2, \ldots, x_n, y_n) = 0$ $(k = 1, 2, \ldots, m)$. To reduce the quantity f using the steepest-descent algorithm while satisfying the constraint equations, the variables x_1, y_1, x_2, y_2, ..., x_n, y_n can be altered according to the equation

$$P^{t+1} = P^t - \mu \left\{ \nabla f + 2 \sum_{k=1}^{m} \left[(\lambda_x)_k \nabla(\mathrm{Re}\, g_k) + (\lambda_y)_k \nabla(\mathrm{Im}\, g_k) \right] \right\}^t ,$$

where the superscripts t and $t+1$ refer to the t^{th} and $(t+1)^{th}$ iterations, respectively, where μ is a scalar controlling the magnitude of the change in the vector P, where the variables $(\lambda_x)_k$ and $(\lambda_y)_k$ are real-valued Lagrangian multipliers corresponding to the real and imaginary parts of the m complex constraint equations, and where the vector P and the operator ∇ are column vectors given by the equations:

$$P = \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ x_n \\ y_n \end{bmatrix} \quad \text{and} \quad \nabla = \begin{bmatrix} \partial/\partial x_1 \\ \partial/\partial y_1 \\ \partial/\partial x_2 \\ \partial/\partial y_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \partial/\partial x_n \\ \partial/\partial y_n \end{bmatrix}$$

The update equation implies that

$$x_j^{t+1} = x_j^t - \mu \left\{ \frac{\partial f}{\partial x_j} + 2 \sum_{k=1}^{m} \left[(\lambda_x)_k \frac{\partial(\mathrm{Re}\, g_k)}{\partial x_j} + (\lambda_y)_k \frac{\partial(\mathrm{Im}\, g_k)}{\partial x_j} \right] \right\}^t$$

and

$$y_j^{t+1} = y_j^t - \mu \left\{ \frac{\partial f}{\partial y_j} + 2 \sum_{k=1}^{m} \left[(\lambda_x)_k \frac{\partial(\mathrm{Re}\, g_k)}{\partial y_j} + (\lambda_y)_k \frac{\partial(\mathrm{Im}\, g_k)}{\partial y_j} \right] \right\}^t$$

Alternatively,

$$(x_j + iy_j)^{t+1} = (x_j + iy_j)^t - \mu \left\{ \left(\frac{\partial f}{\partial x_j} + i \frac{\partial f}{\partial y_j} \right) \right.$$

$$+2 \sum_{k=1}^{m} (\lambda_x)_k \left[\frac{\partial(\mathrm{Re}\, g_k)}{\partial x_j} + i \frac{\partial(\mathrm{Re}\, g_k)}{\partial y_j} \right]$$

$$\left. +2 \sum_{k=1}^{m} (\lambda_y)_k \left[\frac{\partial(\mathrm{Im}\, g_k)}{\partial x_j} + i \frac{\partial(\mathrm{Im}\, g_k)}{\partial y_j} \right] \right\}^t .$$

For linear constraint conditions of the type

$$\sum_{j=1}^{n} \left[(\mathrm{Re}\, V_k)_j - i\,(\mathrm{Im}\, V_k)_j\right] (x_j + iy_j)$$

$$= \sum_{j=1}^{n} \left[x_j(\mathrm{Re}\, V_k)_j + y_j(\mathrm{Im}\, V_k)_j\right] + i\left[y_j(\mathrm{Re}\, V_k)_j - x_j(\mathrm{Im}\, V_k)_j\right]$$

$$= 0,$$

$$\frac{\partial(\mathrm{Re}\, g_k)}{\partial x_j} + i\,\frac{\partial(\mathrm{Re}\, g_k)}{\partial y_j} = (\mathrm{Re}\, V_k)_j + i\,(\mathrm{Im}\, V_k)_j$$

and

$$\frac{\partial(\mathrm{Im}\, g_k)}{\partial x_j} + i\,\frac{\partial(\mathrm{Im}\, g_k)}{\partial y_j} = +i\left[(\mathrm{Re}\, V_k)_j + i(\mathrm{Im}\, V_k)_j\right] .$$

so that

$$(x_j + iy_j)^{t+1} = (x_j + iy_j)^t - \mu\left\{\left(\frac{\partial f}{\partial x_j} + i\,\frac{\partial f}{\partial y_j}\right)\right.$$

$$\left. + 2\sum_{k=1}^{m}\left[(\lambda_x)_k + i\,(\lambda_y)_k\right]\left[(\mathrm{Re}\, V_k)_j + i\,(\mathrm{Im}\, V_k)_j\right]\right\}^t$$

where V_k is an arbitrary complex-valued column vector with n complex-valued components $(\mathrm{Re}\, V_k)_j + i\,(\mathrm{Im}\, V_k)_j$ $(j = 1, 2, \ldots, n)$. Setting $\lambda_k = (\lambda_x)_k + i\,(\lambda_y)_k$ and $(V_k)_j = (\mathrm{Re} V_k)_j + i\,(\mathrm{Im}\, V_k)_j$, the update equation becomes

$$(x_j + iy_j)^{t+1} = (x_j + iy_j)^t - \mu\left\{\left(\frac{\partial f}{\partial x_j} + i\,\frac{\partial f}{\partial y_j}\right) + 2\sum_{k=1}^{m} \lambda_k\, (V_k)_j\right\}^t .$$

The complex-valued Lagrangian multipliers λ_k are determined from the m constraint equations.

2. Single Constraint Mode

Let the output o_n^t at the n^{th} loop of a multiloop frequency-domain adaptive algorithm be given by the formula

$$o_n^t = o_{n-1}^t - (F_n^H)^t X^t ,$$

where o_{n-1}^t is the output of the $(n-1)^{th}$ loop, F_n^H is the n^{th}-loop adaptive filter, and X is the input data vector. The superscript H denotes conjugate transpose. The superscripts t and $t+1$ denote the vectors at iterations t and $t+1$, respectively. From this point on, the superscript t will be understood if the superscript $t+1$ is not used. In the first loop of the adaptive filter, the zero-th loop output is defined to be the beamsteer output $V^H X$, where V^H is the beamsteer filter.

The squared amplitude $o_n o_n^*$ of the n^{th} loop output is

$$o_n o_n^* = o_{n-1} o_{n-1}^* - F_n^H X o_{n-1}^* - o_{n-1} X^H F_n + F_n^H X X^H F_n ,$$

where the superscript $*$ denotes complex conjugate. In the single-constraint mode of the adaptive algorithm, this quantity is reduced subject to the constraint

$$V^H F_n = 0$$

using the steepest-descent algorithm.

Since

$$\frac{\partial(o_n o_n^*)}{\partial(\mathrm{Re}\, F_n)_j} = -2\,\mathrm{Re}\,(X o_{n-1}^*)_j + 2\,\mathrm{Re}\,(X X^H F_n)_j$$

$$\frac{\partial(o_n o_n^*)}{\partial(\mathrm{Im} F_n)_j} = -2\,\mathrm{Im}(X o_{n-1}^*)_j + 2\,\mathrm{Im}(XX^H F_n)_j$$

and

$$\frac{\partial(o_n o_n^*)}{\partial(\mathrm{Re}\, F_n)_j} + i\,\frac{\partial(o_n o_n^*)}{\partial(\mathrm{Im} F_n)_j} = 2(-X o_{n-1}^* + XX^H F_n)_j,$$

where j denotes the j^{th} component of the vectors, the update equation is

$$F_n^{t+1} = F_n + 2\mu\,(X o_{n-1}^* - XX^H F_n - \lambda V)$$

$$= F_n + 2\mu\,(X o_n^* - \lambda V).$$

To determine the Lagrangian multiplier, the relations $V^H F_n^{t+1} = 0$ and $V^H F_n^t = 0$ are used, and the filter update equation is premultiplied by V^H:

$$0 = V^H F_n^{t+1} = V^H F_n^t + 2\mu\,(V^H X o_n^* - V^H V)$$

$$= 2\mu\,(V^H X o_n^* - \lambda V^H V);$$

$$\lambda V^H V = V^H X o_n^*;$$

$$\lambda = \frac{V^H X o_n^*}{V^H V}.$$

Substitution of λ into the update equation yields

$$F_n^{t+1} = F_n + 2\mu\left[X o_n^* - \frac{VV^H X}{V^H V}\, o_n^*\right]$$

$$= F_n + 2\mu\, o_n^* \left[I - \frac{VV^H}{V^H V} \right] X$$

$$= \left[I - \frac{VV^H}{V^H V} \right] \left[F_n + 2\mu\, o_n^* X \right] .$$

In the last step, the fact that $V^H F_n = 0$ was used.

3. Multiple Constraint Mode

In the multiple-constraint mode, the output o_n^t at the n^{th} loop of the algorithm is again

$$o_n^t = o_{n-1}^t - (F_n^H)^t X^t ,$$

but the squared output $o_n o_n^*$ is reduced subject to the multiple constraints

$$F_{k-1}^H F_n = 0 \qquad (k = 1, 2, \ldots, n),$$

where F_o^H is the beamsteer filter. Therefore the update equation is

$$F_n^{t+1} = F_n + 2\mu\, (X o_n^* - \sum_{k=1}^{n} F_{k-1} \lambda_{k-1}) .$$

If the matrix M and the column vector Λ are defined by the equations

$$M = \begin{bmatrix} V^H \\ F_1^H \\ \vdots \\ F_{n-1}^H \end{bmatrix} \text{ and } \Lambda = \begin{bmatrix} \lambda_o \\ \lambda_1 \\ \vdots \\ \lambda_{n-1} \end{bmatrix} ,$$

the update equation may be rewritten

$$F_n^{t+1} = F_n + 2\mu\,(X o_n^* - M^H \Lambda).$$

To determine the column vector Λ of Lagrangian multipliers, the update equation is premultiplied by M^{t+1}:

$$O = M^{t+1} F_n^{t+1}$$

$$= M^{t+1} F_n + 2\mu\,(o_n^* M^{t+1} X - M^{t+1} M^H \Lambda)\ ,$$

where O is a column vector with n zeroes;

$$2\mu\, M^{t+1} M^H \Lambda = M^{t+1} F_n + 2\mu\, o_n^* M^{t+1} X \ ;$$

$$\Lambda = \frac{1}{2\mu}\,(M^{t+1} M^H)^{-1} M^{t+1} F_n + o_n^* (M^{t+1} M^H)^{-1} M^{t+1} X \ .$$

Substituting Λ into the update equation,

$$F_n^{t+1} = F_n + 2\mu\, o_n^* X$$

$$- M^H (M^{t+1} M^H)^{-1} M^{t+1} F_n - 2\mu\, o_n^* M^H (M^{t+1} M^H)^{-1} M^{t+1} X$$

$$= \left\{ I - (M^H)^t \left[M^{t+1} (M^H)^t \right]^{-1} M^{t+1} \right\} \left[F_n^t + 2\mu\,(o_n^*)^t X^t \right].$$

In the second loop,

$$M^{t+1} (M^H)^t = \begin{bmatrix} V^H \\ (F_1^H)^{t+1} \end{bmatrix} \begin{bmatrix} V & F_1^t \end{bmatrix}$$

$$= \begin{bmatrix} V^H V & V^H F_1^t \\ (F_1^H)^{t+1} V & (F_1^H)^{t+1} F_1^t \end{bmatrix}$$

$$= \begin{bmatrix} V^H V & 0 \\ 0 & (F_1^H)^{t+1} F_1^t \end{bmatrix} ,$$

so that

$$\left[M^{t+1} (M^H)^t\right]^{-1} = \begin{bmatrix} \dfrac{1}{V^H V} & 0 \\ 0 & \dfrac{1}{(F_1^H)^{t+1} F_1^t} \end{bmatrix}$$

and

$$(M^H)^t \left[M^{t+1} (M^H)^t\right]^{-1} M$$

$$= \begin{bmatrix} V & F_1^t \end{bmatrix} \begin{bmatrix} \dfrac{1}{V^H V} & 0 \\ 0 & \dfrac{1}{(F_1^H)^{t+1} F_1^t} \end{bmatrix} \begin{bmatrix} V^H \\ (F_1^H)^{t+1} \end{bmatrix}$$

$$= \begin{bmatrix} \dfrac{V}{V^H V} & \dfrac{F_1^t}{(F_1^H)^{t+1} F_1^t} \end{bmatrix} \begin{bmatrix} V^H \\ (F_1^H)^{t+1} \end{bmatrix}$$

$$= \frac{V V^H}{V^H V} + \frac{F_1^t (F_1^H)^{t+1}}{(F_1^H)^{t+1} F_1^t} .$$

Thus the second-loop update equation is

$$F_2^{t+1} = \left[I - \frac{VV^H}{V^H V} - \frac{F_1^t \,(F_1^H)^{t+1}}{(F_1^H)^{t+1} F_1^t} \right] \left[F_2^t + 2\mu \,(o_2^*)^t X^t \right]$$

B. MULTILOOP ADAPTIVE FILTERING - TIME DOMAIN

1. Single Constraint Mode

Let the output o_n^t at the n^{th} loop of a multiloop time-domain adaptive algorithm be

$$o_n^t = o_{n-1}^t - (F_n^T)^t X^t ,$$

where o_{n-1}^t is the output of the $(n-1)^{th}$ loop, F_n^T is the n^{th}-loop adaptive filter, and X is the input data vector. The superscript T denotes transpose. In the first loop of the adaptive filter, the zero-th loop output is defined to be the beamsteer output $V^T X$, where V^T is the beamsteer filter.

The squared output o_n^2 of the n^{th} loop output is

$$o_n^2 = o_{n-1}^2 - F_n^T X o_{n-1} - o_{n-1} X^T F_n + F_n^T X X^T F_n .$$

In the "single-constraint" mode of the adaptive algorithm, the squared output o_n^2 is reduced subject to the non-redundant constraints expressed in the matrix equation

$$MF_n = O$$

using the steepest-descent algorithm. Here M is a matrix with as many rows as constraint conditions and as many columns as filter weights, F_n is a column vector containing the filter weights, and O is an all-zero column vector with

as many elements as constraint conditions. Several constraints were permitted so that the frequency response to a signal from the desired look direction for the combined filter

$$V^T - \sum_{k=1}^{n} F_k^T$$

could be made white.

The update equation for the adaptive algorithm is

$$F_n^{t+1} = F_n - \mu \nabla \left[o_n^2 + 2\Lambda^T MF_n \right]$$

$$= F_n + 2\mu \left[(o_{n-1} - X^T F) X - M^T \Lambda \right]$$

$$= F_n + 2\mu \left[o_n X - M^T \Lambda \right]$$

where ∇ denotes a column vector operator containing partial derivatives with respect to the corresponding filter weights in F_n^t, and where Λ is a column vector having as elements Lagrangian multipliers corresponding to each of the constraint conditions. The constraint conditions $MF_n^{t+1} = O$ and $MF_n^t = O$ are used to determine the column vector Λ by premultiplying the update equation by M :

$$O = MF_n^{t+1} = MF_n + 2\mu \left[o_n MX - MM^T \Lambda \right]$$

$$= 2\mu \left[o_n MX - MM^T \Lambda \right] ;$$

$$MM^T \Lambda = o_n MX ;$$

$$\Lambda = o_n (MM^T)^{-1} MX.$$

Substitution into the update equation gives

$$\begin{aligned} F_n^{t+1} &= F_n + 2\mu\, o_n \left[X - M^T (MM^T)^{-1} MX \right] \\ &= F_n + 2\mu\, o_n \left[I - M^T (MM^T)^{-1} M \right] X \\ &= \left[I - M^T (MM^T)^{-1} M \right] \left[F_n + 2\mu\, o_n X \right]. \end{aligned}$$

2. Multiple Constraint Mode

The output o_n^t at the n^{th} loop of the algorithm is the same as for the "single-constraint" mode, but the constraints

$$F_k^T F_n = 0 \qquad (k = 1, 2, \ldots, n-1)$$

supplement those expressed in the matrix equation

$$M_o F_n = O_c ,$$

where M_o is a time-invariant matrix used to express c linear constraint conditions and O_c is a column vector with c zeroes.

The update equation is

$$\begin{aligned} F_n^{t+1} &= F_n - \mu \nabla \left[o_n^2 + 2\, \Lambda_o^T M_o F_n + \sum_{k=1}^{n-1} 2 \lambda_k F_k^T F_n \right] \\ &= F_n + 2\mu \left(o_n X - M_o^T \Lambda_o - \sum_{k=1}^{n-1} F_k \lambda_k \right) , \end{aligned}$$

where Λ_o is a column vector containing Lagrangian multipliers corresponding to the c constraint conditions expressed in the matrix equation $M_o F_n = O_c$, and where the values λ_k are scalar Lagrangian multipliers corresponding to the new constraint conditions.

If a new matrix M and column vector Λ are defined by the equation

$$M = \begin{bmatrix} M_o \\ F_1^T \\ \vdots \\ F_{n-1}^T \end{bmatrix} \quad \text{and } \Lambda = \begin{bmatrix} \Lambda_o \\ \lambda_1 \\ \vdots \\ \lambda_{n-1} \end{bmatrix} ,$$

the update equation may be rewritten

$$F_n^{t+1} = F_n + 2 \; (o_n X - M^T \Lambda) .$$

M now has $c + n-1$ rows, and Λ now has $c + n-1$ elements.

To find the column vector Λ , the update equation is pre-multiplied by M^{t+1}:

$$O_{c+n-1} = M^{t+1} F_n^{t+1}$$

$$= M^{t+1} F_n + 2\mu (o_n M^{t+1} X - M^{t+1} M^T \Lambda) .$$

O_{c+n-1} is a column vector with $c+n-1$ zeroes. The superscript $t+1$ is attached to the matrix M because it contains the time varying filters $F_1^T, \ldots,$ F_{n-1}^T. Rearrangement of the preceding equation gives the formula

$$2\mu M^{t+1} M^T \Lambda = M^{t+1} F_n + 2\mu o_n M^{t+1} X$$

and hence the result

$$\Lambda = \frac{1}{2\mu} \; (M^{t+1} M^T)^{-1} M^{t+1} F_n + o_n (M^{t+1} M^T)^{-1} M^{t+1} X .$$

Incorporating this result in the update equation,

$$\begin{aligned} F_n^{t+1} &= F_n + 2\mu o_n X \\ &\quad - M^T(M^{t+1}M^T)^{-1}M^{t+1}F_n - 2\mu o_n M(M^{t+1}M^T)^{-1}M^{t+1}X \\ &= \left\{ I - (M^T)^t \left[M^{t+1}(M^T)^t \right]^{-1} M^{t+1} \right\} (F_n^t + 2\mu o_n^t X^t) \, . \end{aligned}$$

In the second loop,

$$\begin{aligned} M^{t+1}(M^T)^t &= \left[\begin{array}{c} M_o \\ \hline (F_1^T)^{t+1} \end{array} \right] \left[\begin{array}{c|c} M_o^T & F_1^t \end{array} \right] \\ &= \left[\begin{array}{c|c} M_o M^T & M_o F_1^t \\ \hline (F_1^T)^{t+1} M_o^T & (F_1^T)^{t+1} F_1^T \end{array} \right] \\ &= \left[\begin{array}{c|c} M_o M_o^T & O_c \\ \hline O_c^T & (F_1^T)^{t+1} F_1^t \end{array} \right] . \end{aligned}$$

The inverse of $M^{t+1}(M^T)^t$ is

$$\left[\begin{array}{c|c} (M_o M_o^T)^{-1} & O_c \\ \hline O_c^T & \dfrac{1}{(F_1^T)^{t+1} F_1^t} \end{array} \right]$$

and

$$\begin{aligned} &(M^T)^t \left[M^{t+1}(M^T)^t \right]^{-1} M^{t+1} \\ &= \left[\begin{array}{c|c} M_o^T & F_1^t \end{array} \right] \left[\begin{array}{c|c} (M_o M_o^T)^{-1} & O_c \\ \hline O_c^T & \dfrac{1}{(F_1^T)^{t+1} F_1^t} \end{array} \right] \left[\begin{array}{c} M_o \\ \hline (F_1^T)^{t+1} \end{array} \right] \end{aligned}$$

$$= \left[M_o^T(M_o M_o^T)^{-1} \;\middle|\; \frac{F_1^T}{(F_1^T)^{t+1} F_1^t} \right] \left[\begin{array}{c} M_o \\ \hdashline (F_1^T)^{t+1} \end{array} \right]$$

$$= M_o^T(M_o M_o^T)^{-1} M_o + \frac{F_1^t (F_1^T)^{t+1}}{(F_1^T)^{t+1} F_1^t} \, .$$

Hence the second-loop update equation is

$$F_2^{t+1} = \left[I - M_o^T(M_o M_o^T)^{-1} M_o - \frac{F_1^t (F_1^T)^{t+1}}{(F_1^T)^{t+1} F_1^t} \right] \left[F_2^t + 2\mu o_2^t X^t \right] .$$

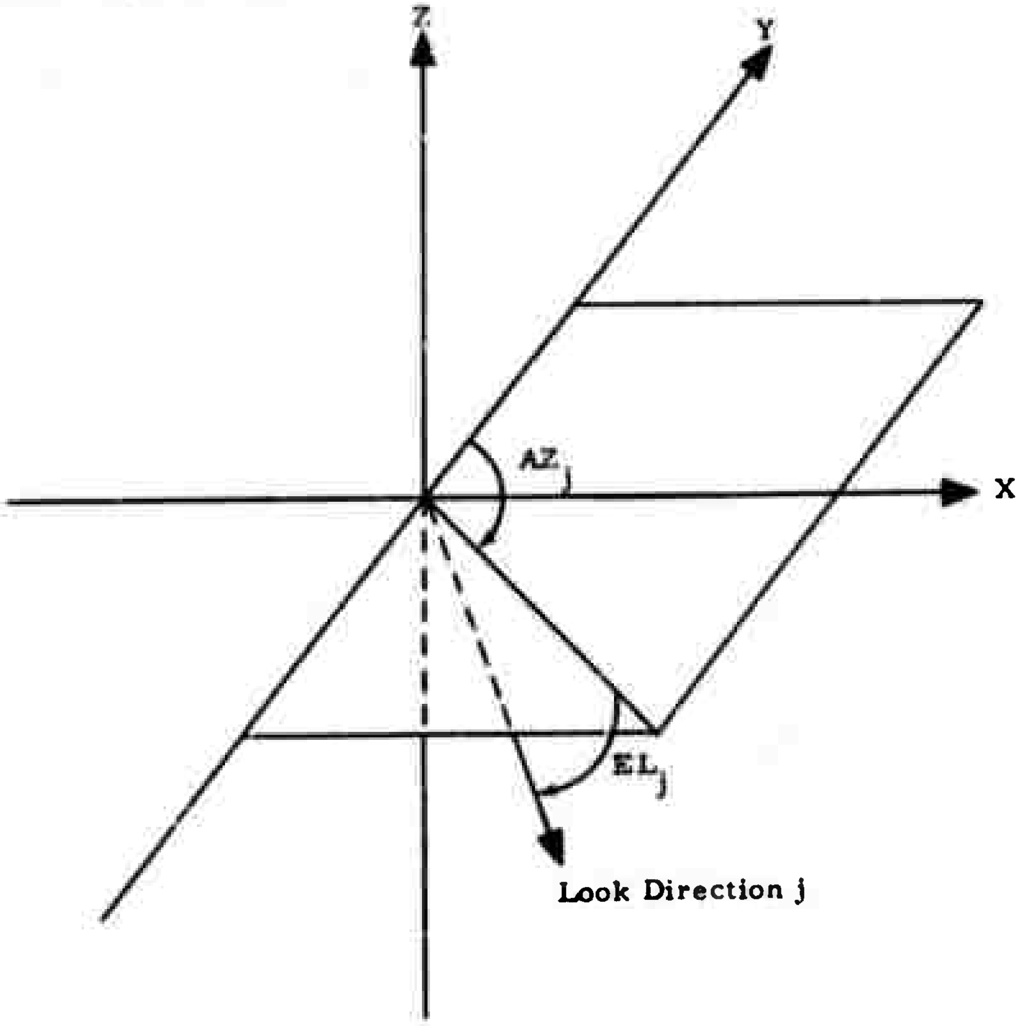
C. DATA GENERATION

An array of elements located in a three dimensional space is defined by a set of coordinates, (X_k, Y_k, Z_k) where $k = 1, 2, \ldots, C$, and C is the number of elements in the array. Signal vectors in the frequency domain and at frequency f, S_j, for the array are defined for each of the specified "look directions" by the equation:

$$S_j = \begin{bmatrix} e^{-i2\pi f \Delta t_j(1)} \\ e^{-i2\pi f \Delta t_j(2)} \\ \vdots \\ e^{-i2\pi f \Delta t_j(C)} \end{bmatrix}$$

where j references a specific "look direction". The j^{th} "look direction" is defined by a set of angles AZ_j (Azimuth) and EL_j (Elevation) as illustrated below.

The $\Delta t_j(k)$ is the relative time delay of the signal at array element k and is dependent upon the "look direction", j, and the propagation velocity of the signal, VEL.



Consequently S_j can also be defined as

$$S_j = \begin{bmatrix} e^{-i2\pi f \frac{\vec{U}\cdot\vec{R}(1)}{VEL}} \\ e^{-i2\pi f \frac{\vec{U}\cdot\vec{R}(2)}{VEL}} \\ \vdots \\ e^{-i2\pi f \frac{\vec{U}\cdot\vec{R}(C)}{VEL}} \end{bmatrix}$$

where $\vec{R}$ (k) is a vector in the array coordinate space defining

the location of element k relative to some reference element, and $\vec{U}$ is a unit vector from the reference element in the direction of the "look direction". Therefore, the components of $\vec{U}$ are:

$$U_x = \text{Cos}\,(EL)\,\text{Sin}\,(AZ)$$

$$U_y = \text{Cos}\,(EL)\,\text{Cos}\,(AZ)$$

$$U_z = \text{Sin}\,(EL)$$

A composite cross-power matrix, Ω, is generated from p signal vectors plus random noise according to the following formula:

$$\Omega = \sum_{j=1}^{p} \alpha_j S_j S_j^H + \beta I \text{ ---------------------------- } (1)$$

where α_j and β are the relative weights to be associated with the signal vectors and random noise component, respectively.

If we let x be a complex random vector such that

$$x = \sum_{j=1}^{p} \epsilon_j S_j + \delta w \text{ ---------------------------- } (2)$$

where ϵ_j is a random number with normal distribution, zero mean and variance α_j, and where

w is a complex random vector such that $\overline{ww^H} = I$, then by choosing

$$\delta = \sqrt{\beta}$$

it follows that

$$\Omega_x = \overline{XX^H} = \overline{\left(\sum_{j=1}^{p} \epsilon_j S_j + \delta w\right)\left(\sum_{j=1}^{p} \epsilon_j S_j + \delta w\right)^H}$$

$$= \sum_{j=1}^{p} \overline{\epsilon_j^2} S_j S_j^H + \delta^2 \overline{ww^H}$$

$$= \sum_{j=1}^{p} \alpha_j S_j S_j^H + \beta I$$

Thus, we can use equation (2) to generate a set of data vectors which has the crosspower matrix Ω in equation (1).

D. EVALUATION AND DISPLAY

We try to evaluate the performance of the multiloop adaptive filtering technique by comparing its mean square output (MSO) at time t for each of the signal vectors, white noise and composite energy with those of the optimum maximum-likelihood filter.

The formula we used to calculate the MSO is as follows:

	Optimum Filter	Adaptive Filter Loop K
White Noise	$\beta F_o^H F_o$	$\beta(V - \sum_{i=1}^{K} F_i^t)^H (V - \sum_{i=1}^{K} F_i^t)$
Signal Vector j	$\alpha_j \left\| F_o^H S_j \right\|^2$	$\alpha_j \left\| (V - \sum_{i=1}^{K} F_i^t)^H S_j \right\|^2$
Composite Energy	$F_o^H \Omega F_o$	$(V - \sum_{i=1}^{K} F_i^t)^H \Omega (V - \sum_{i=1}^{K} F_i^t)$

where F_o is the maximum-likelihood filter and

$$F_o = \frac{Z}{S_1^H Z}$$

where $Z = \Omega^{-1} S_1$

and S_1 is the signal vector to be preserved.

We take the ratio of optimum filter MSO to adaptive filter loop k MSO and plot it as a function of t (number of adaptive iterations) for each of the signal vectors, white noise and composite energy, respectively.

SECTION III
PROCESSING RESULTS AND CONCLUSIONS

A set of synthetic data was generated for an equally-spaced line array of five elements. Three coherent components were added to synthetic random noise. The first of the three coherent components, S_1, was a signal from the look direction. The second and third coherent components, S_2 and S_3, were interfering noise components. Different weightings were assigned to these vectors (as shown in Figure III-1). The weighting $\alpha_1 = 0$ for the signal component S_1 indicates that the signal component was not considered to be part of the noise output. Component 3 represented a weak noise component in the presence of a strong noise component (component 2). It was expected that multiloop adaptive filtering would increase the sensitivity of adaptive processing to the weak noise component.

Several runs with two-loop adaptive filters were made using 500 synthetically-generated data vectors as input to a like number of iterations of the adaptive-filter algorithm. Using the frequency-domain algorithms derived in the previous section, the adaptive-filter algorithm was run both in the single-constraint mode and the multiple-constraint mode. Two different sets of initial filters were tried. The first was an optimum filter designed for the data as specified, except that component 3 was omitted. The second was a simple beamsteer filter.

The curves plotted are the ratios

$$F_o^H \, \Omega_c \, F_o \Big/ F_A^H \, \Omega \, F_A ,$$

where Ω_c is the crosspower spectrum matrix for each of the noise components, F_o^H is the optimum filter and F_A^H is the composite adaptive filter

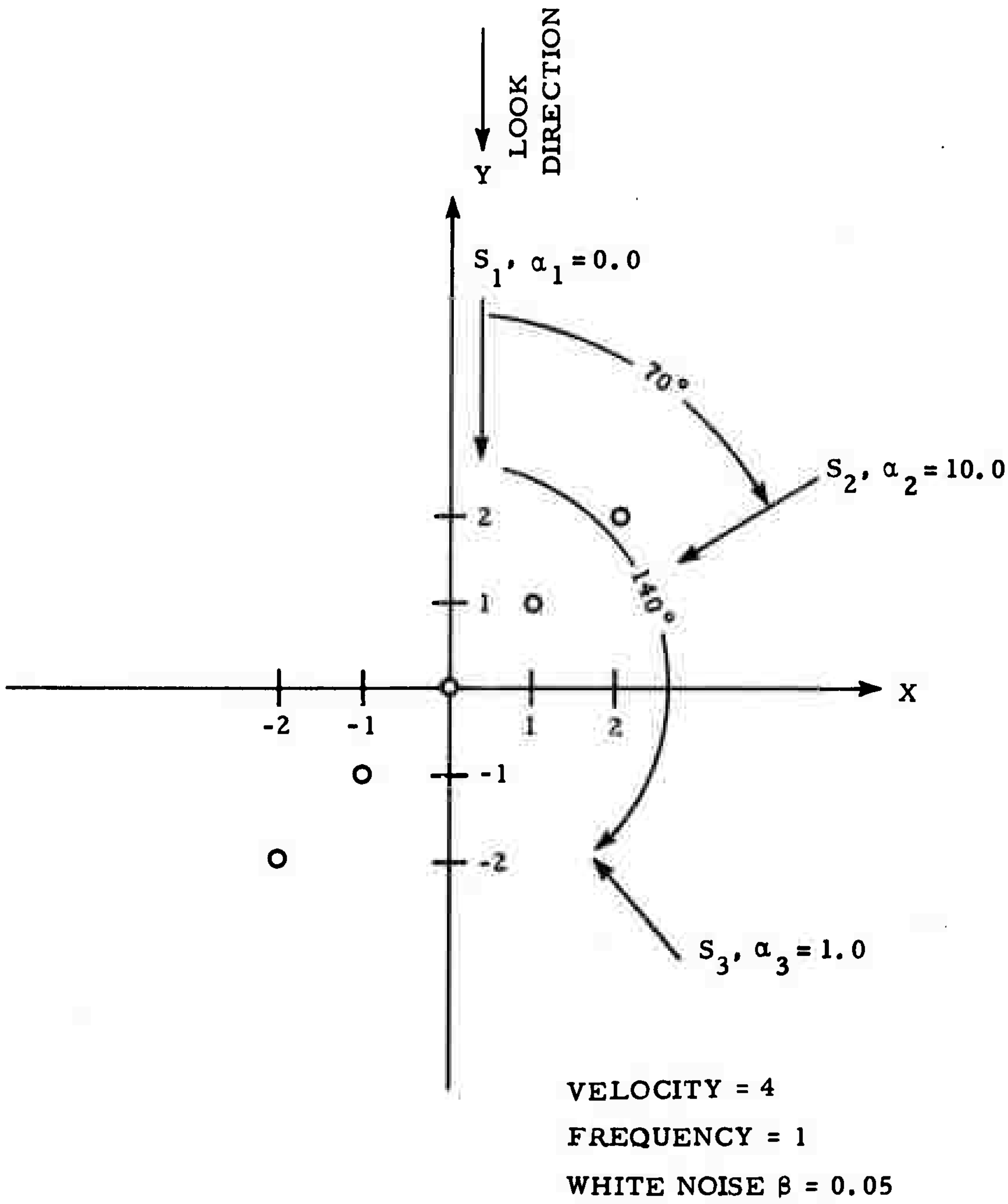


Figure III-1. Data Model

vector corresponding to the iteration for which the power ratios are plotted. Each curve, therefore, measures the performance of the adaptive filter relative to the optimum filter for each noise component or all noise components combined.

Figures III-2 and III-3 show the single-constraint-mode results for the one-loop and two-loop algorithms through 500 iterations. The starting filter was an optimum filter designed for the specified noise field buth with component 3 omitted. Convergence rates were $\mu_1 = 0.00015$ in the first loop and $\mu_2 = 0.0003$ in the second loop. The two-loop algorithm concentrated much more effort on the weak component S_3 and actually suppressed this energy more than the optimum filter. The results of Figure III-3 reflect the single-constraint mode of processing, where the second-loop filter is constrained only to pass the signal.

In the top half of Figure III-4 are presented the numerical values of the optimum filter set. Differences between the optimum filter set and the one-loop or two-loop filter sets after 500 iterations are shown in the bottom half. For purposes of comparison, the differences between the optimum filter weights and the initial filter weights are also shown in the bottom half of the figure. The circles representing the two-loop system lie much closer to zero than the triangles corresponding to the one-loop system. Thus the two-loop system appears to have converged significantly faster toward the optimum filter.

A similar set of results is shown in Figures III-5 and III-6. In this case, the convergence rates were increased and the beamsteer filter set was used as the initial filter set. Again, the two-loop algorithm appears to have converged more rapidly toward the optimum filter and to have responded more rapidly to the weaker noise component.

Figure III-7 gives the multiple-constraint results for the two-loop algorithm after 500 iterations. It can be compared with Figure III-3 to

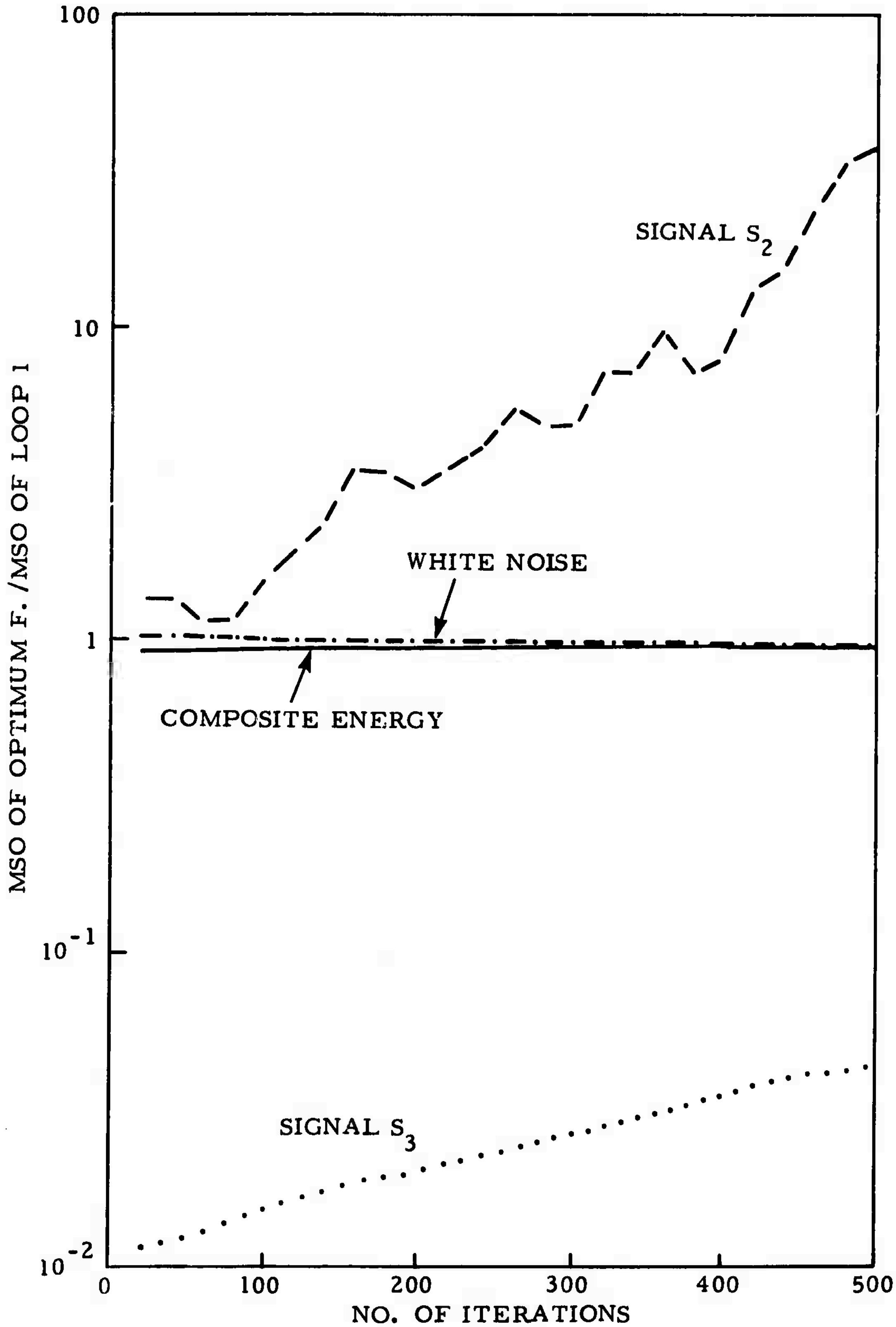


Figure III-2. Single Constraint Mode - One Loop Starting with Optimum Filter Designed without S_3(μ_1 = .00015)

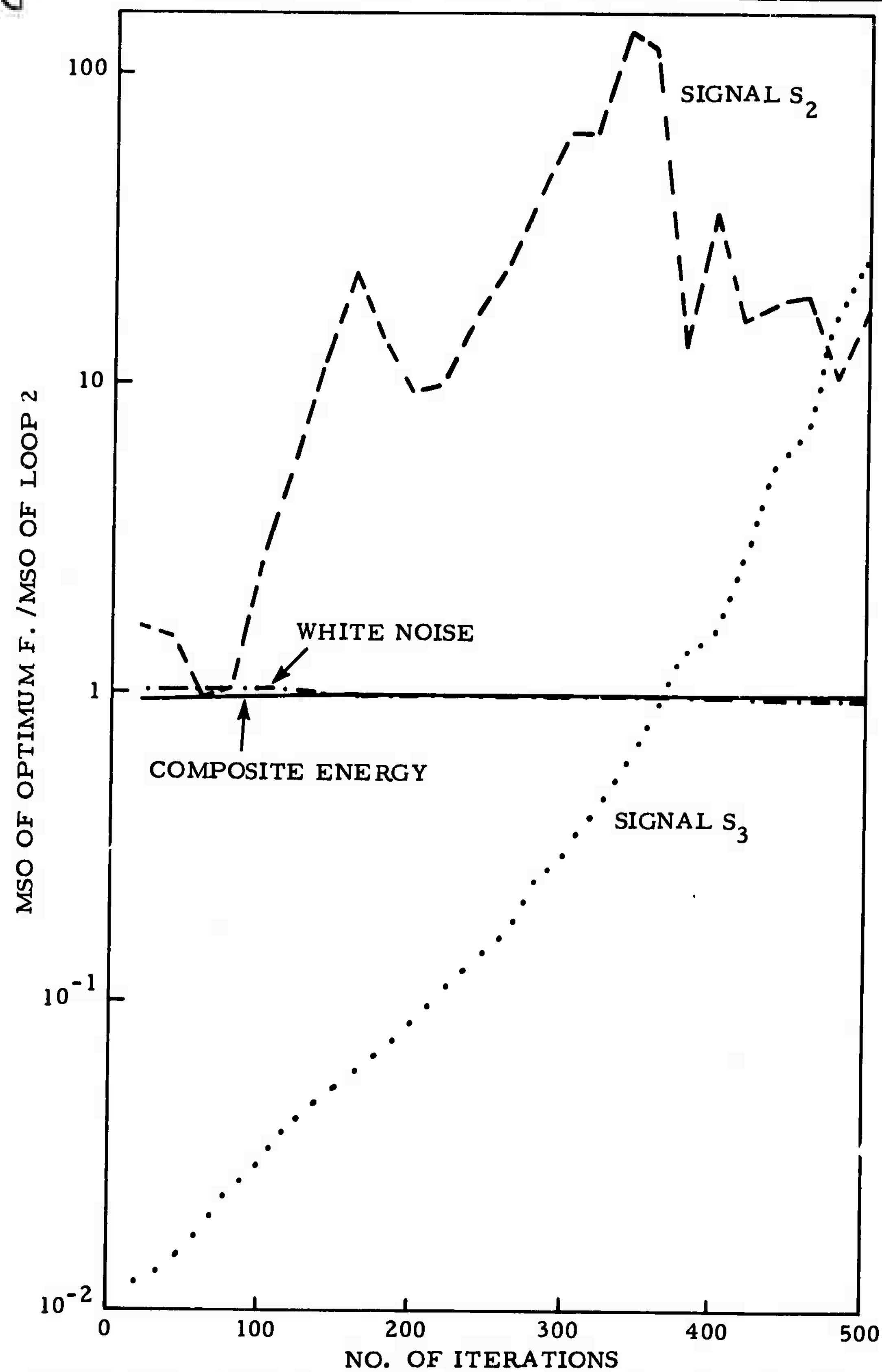


Figure III-3. Single Constraint Mode - Two Loops Starting with Optimum Filter Designed without S_3(μ_1 = .00015, μ_2 = .0003)

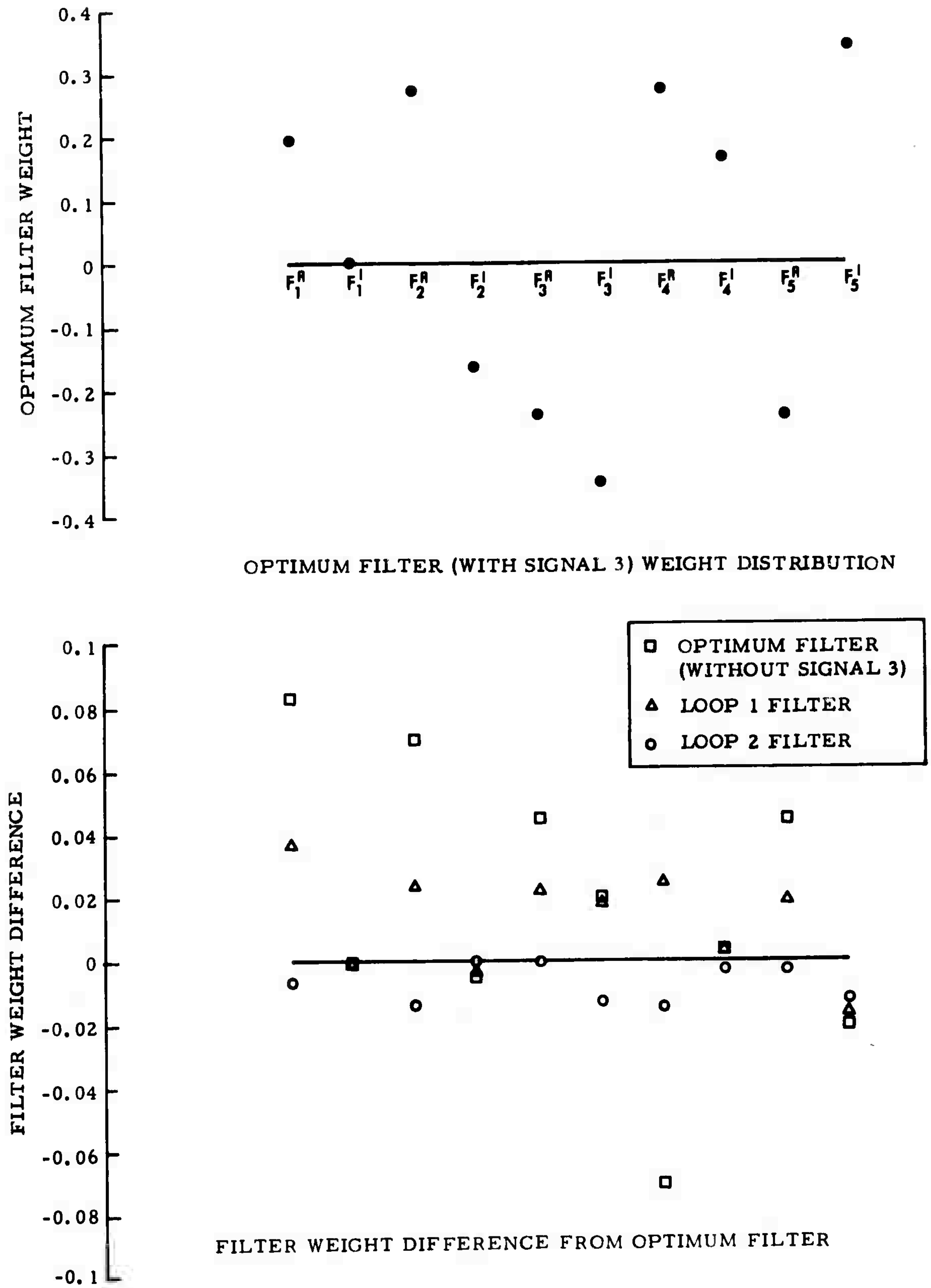


Figure III-4. Filter Coefficients

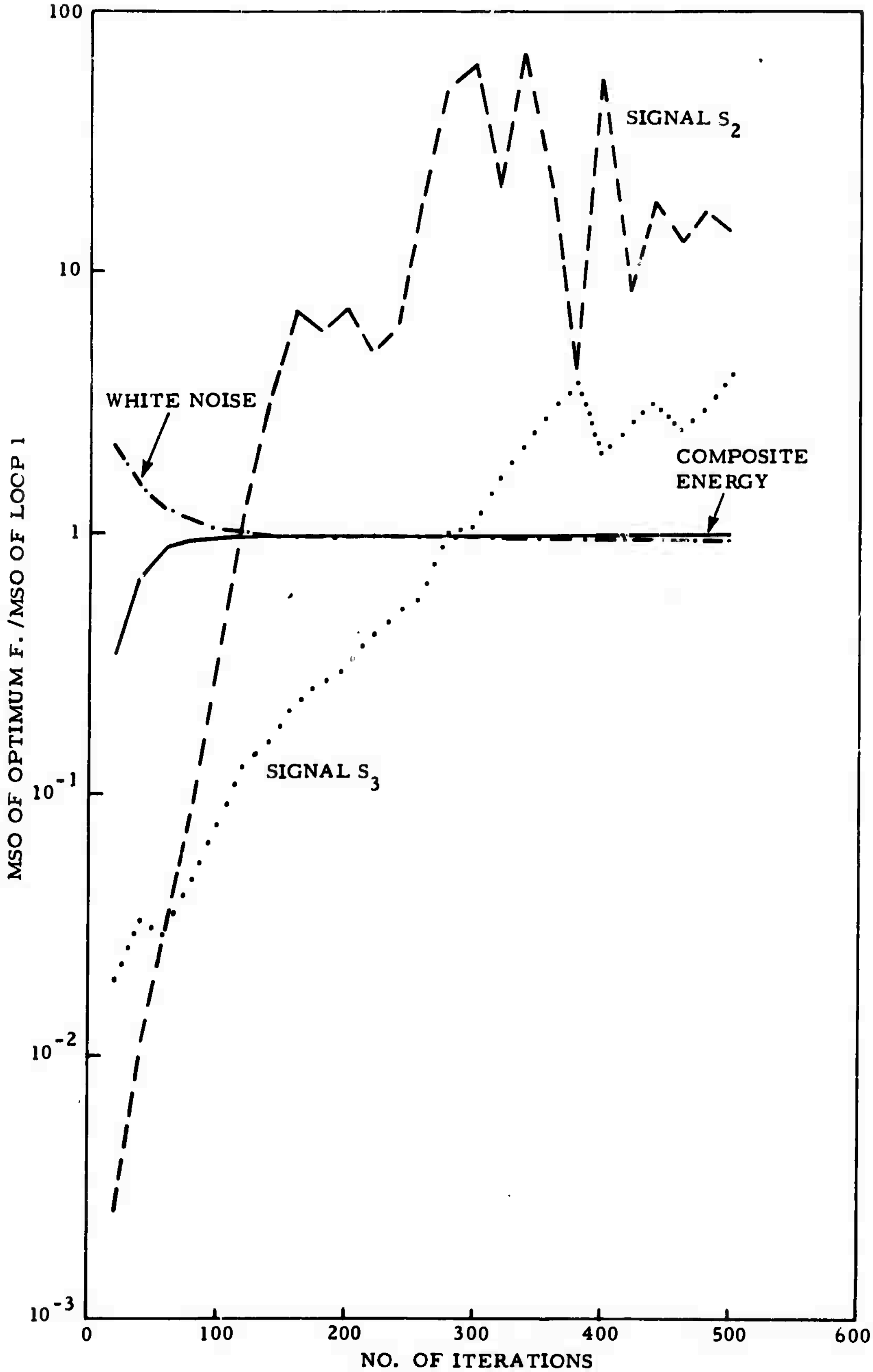


Figure III-5. Single Constraint Mode - One Loop Starting with Beamsteer Filter (μ_1 = .001)

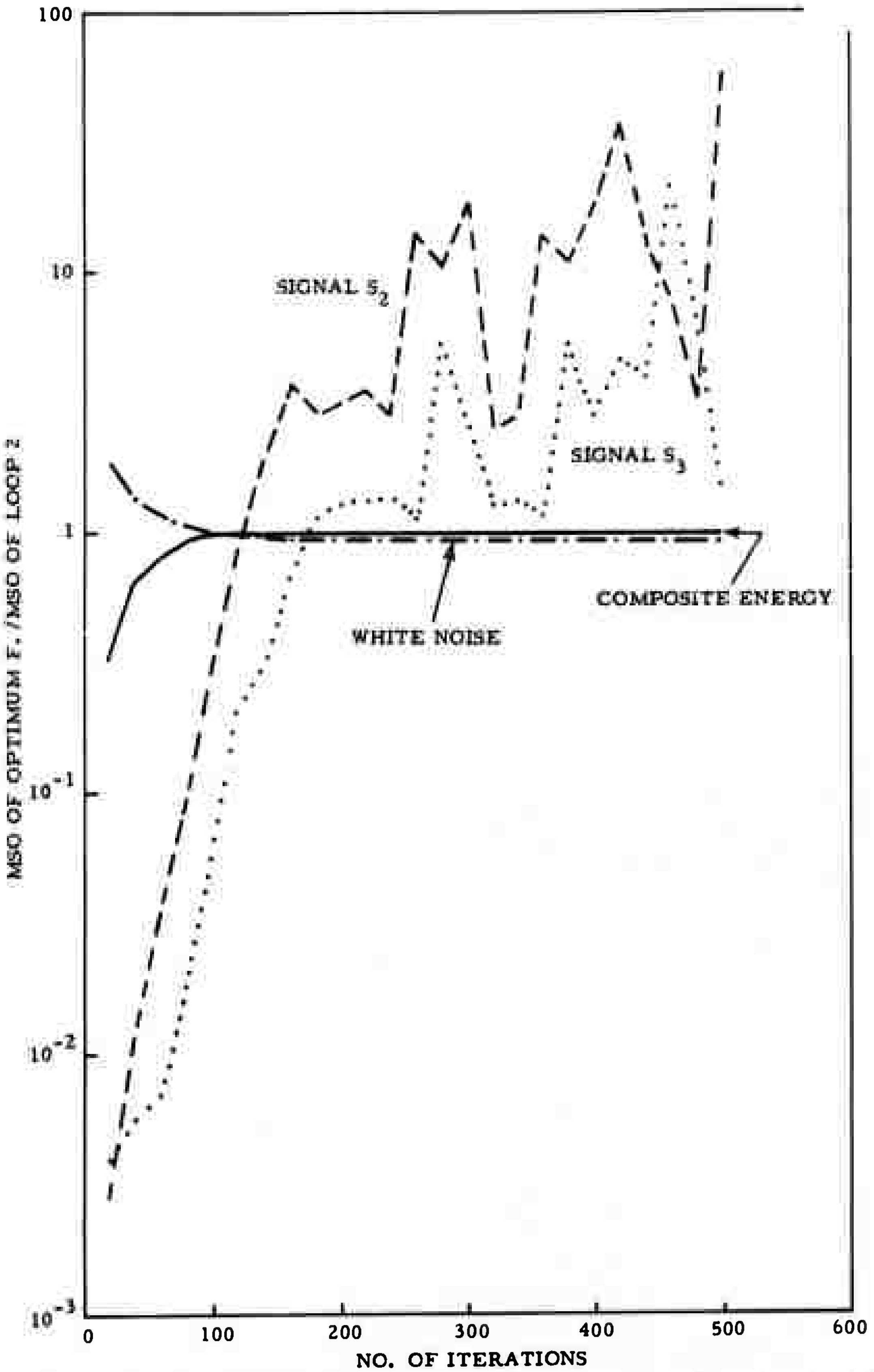


Figure III-6. Multiple Constraint Mode - Two Loops Starting with Beamsteer Filter (μ_1 = .001, μ_2 = .0015)

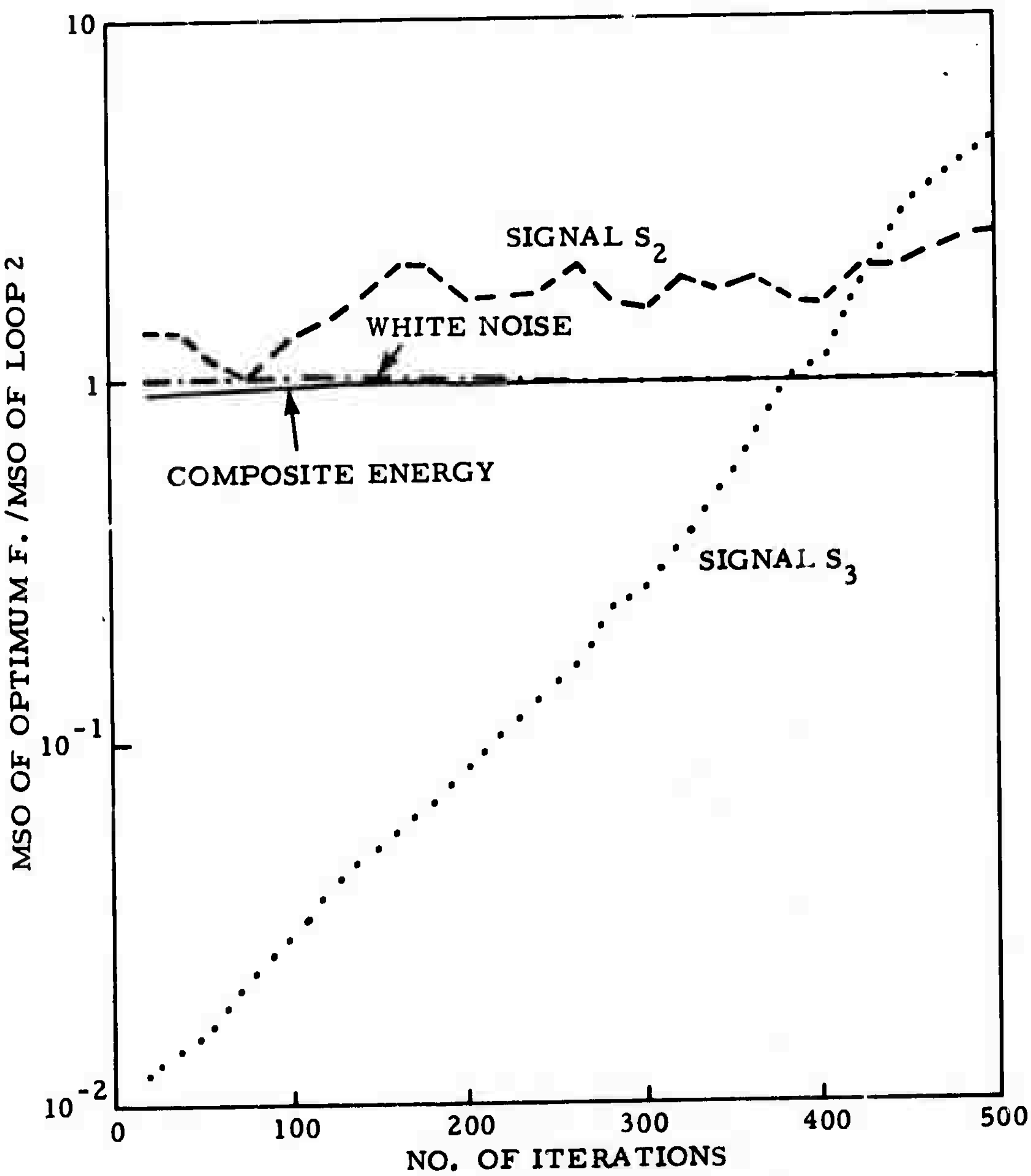


Figure III-7. Multiple Constraint Mode - Two Loops Starting with Optimum Filter Designed without S_3 (μ_1 = .00015, μ_2 = .0003)

determine the effect of the constraint requiring orthogonality between F_1 in the first loop and F_2 in the second loop. Both figures depict results obtained from an initial filter designed to minimize the specified noise field (with the weaker coherent noise component removed). Convergence rates in both cases were $\mu_1 = 0.00015$ in the first loop and $\mu_2 = 0.0003$ in the second loop. The orthogonality constraint appears to have stabilized the convergence process. The noise output power for the individual components did not fluctuate nearly as much in the multiple-constraint mode. Furthermore, the noise output power for the various noise components after 500 iterations was much closer to that which would have been obtained from an optimum system.

It is interesting that the various adaptive algorithms perform better than the optimum filter on the discrete coherent noise components at the expense of poorer performance on the white noise and composite noise.

The chief objective of this study was to determine whether multiloop adaptive filtering could reduce the differences in the rate of convergence associated with unequal eigenvalues in the noise matrix. The simulation results have demonstrated that multiloop adaptive filtering does indeed have this capability. This capability is of potential use in speeding the rate of convergence when the noise statistics vary slowly with time. In the event of such quasi-stationary noise statistics, the single-loop adaptive algorithm would otherwise tend to hinder adjustment to the changing noise statistics because of unequal eigenvalues in the noise matrix.

SECTION IV

REFERENCES

L. E. Brennan, et al., "Control Loop Noise in Adaptive Array Antennas," IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-7, No. 2, March 1971.